

BATCH BAYESIAN ACTIVE LEARNING FOR FEASIBLE REGION IDENTIFICATION BY LOCAL PENALIZATION

Jixiang Qing
Nicolas Knudde
Ivo Couckuyt
Tom Dhaene

Department of Information Technology
Ghent University - imec
Technologiepark-Zwijnaarde 126
Ghent, 9052, BELGIUM

Kohei Shintani

Vehicle Engineering Development Div
Toyota Motor Corporation
Aichi, JAPAN

ABSTRACT

Identifying all designs satisfying a set of constraints is an important part of the engineering design process. With physics-based simulation codes, evaluating the constraints becomes considerable expensive. Active learning can provide an elegant approach to efficiently characterize the feasible region, i.e., the set of feasible designs. Although active learning strategies have been proposed for this task, most of them are dealing with adding just one sample per iteration as opposed to selecting multiple samples per iteration, also known as batch active learning. While this is efficient with respect to the amount of information gained per iteration, it neglects available computation resources. We propose a batch Bayesian active learning technique for feasible region identification by assuming that the constraint function is Lipschitz continuous. In addition, we extend current state-of-the-art batch methods to also handle feasible region identification. Experiments show better performance of the proposed method than the extended batch methods.

1 INTRODUCTION

In many real-world problems, we wish to identify all designs satisfying a set of constraints. These feasible designs can be further analyzed, optimized, and refined in the design pipeline. This problem is common in engineering applications such as structure reliability analysis (Dubourg et al. 2013), factory optimization analysis (Bryan et al. 2006), environmental monitoring (Gotovos et al. 2013) and flight vehicle designs (Larson and Mattson 2012). In those problem settings, often the feasibility can be determined directly through (physics-based) simulations or measurements. However, such evaluations generally require considerable time and budget.

Metamodeling (also known as surrogate modeling) is a popular approach to limit the number of evaluations. A metamodel provides a compact approximation of the expensive simulator. Accordingly, metamodels and sequentially sampling techniques, also known as active learning, are useful to efficiently identify feasible designs.

Active Learning (AL) policies try to select the most informative observations for updating an existing belief. For instance, Bayesian optimization (BO) (Shahriari et al. 2015) is typically used to find the global optimum of an unknown function (Hernández-Lobato et al. 2014).

More formally, in feasible region identification problem, the aim is to accurately predict whether a given unsimulated candidate \mathbf{x}_* is feasible, i.e., if its outcome $f(\mathbf{x}_*)$ is in a pre-specified range $[\alpha, \beta]$. The region where all candidates are feasible is denoted as *level set* (Gotovos et al. 2013), *excursion set*

(Azzimonti et al. 2019), *feasible domain* (Chen and Fuge 2018) or *feasible region* (Knudde et al. 2019). In this paper we follow the notion of (Knudde et al. 2019) by defining the feasible region A :

$$A := \{\mathbf{x} \in \mathcal{X} : \alpha \leq f(\mathbf{x}) \leq \beta\}$$

Since a candidate can only be either feasible or infeasible, its feasibility can alternatively be represented by an indicator function: $I_A : \mathcal{X} \rightarrow \{0, 1\}$.

$$I_A(f(\mathbf{x})) = \begin{cases} 1 & \alpha \leq f(\mathbf{x}) \leq \beta \\ 0 & \text{otherwise} \end{cases}$$

Without loss of generality, $f : \mathcal{X} \rightarrow \mathbb{R}$ is an unknown function defined on a compact subset $\mathcal{X} \subset \mathbb{R}^d$. In particular, the focus is on the case where f is deterministic, although a stochastic example is also considered in Section 4.

Given the training set $D_n = \{\mathbf{X}_n, \mathbf{y}_n\}$, two Bayesian machine learning frameworks can be utilized. If the values in \mathbf{y}_n are discrete (e.g., Snoek 2013), the problem is a classification problem. In this work we are concerned with a continuous f . It can still be treated as a classification problem by applying the indicator function to f specifying if constraints have been satisfied (Chen and Fuge 2018). However, it makes more sense to use f directly if possible as it entails more information about the problem. The key reasoning behind this approach is that more information can be retrieved from the response of f , as continuous observations also provide a clue about how close the constraint boundary is (Gelbart 2015), and, hence, can also be exploited by AL strategies (Bryan et al. 2006, Picheny et al. 2010, Knudde et al. 2019).

An appropriate metamodel to approximate the continuous function f with limited observations is Kriging (Forrester et al. 2008, Van Steenkiste et al. 2016, Rojas-Gonzalez et al. 2019). Kriging is also known as a Gaussian Process (GP) (Rasmussen 2003): $p(f) = \text{GP}(\mu, k)$, where $\mu : \mathcal{X} \rightarrow \mathbb{R}$ (assumed zero in this research) is the mean function, and $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is the kernel. Assuming Gaussian noise on the observations: $p(y|f) \sim \mathcal{N}(0, \sigma_n^2)$, the posterior distribution of the latent function at a test point \mathbf{x}_* : $p(f_*|D_n, \mathbf{x}_*)$ is also Gaussian,

$$\mathcal{N}(K_{*n}^T(K_{nn} + \sigma_n^2 I)^{-1} \mathbf{y}, K_{**} - K_{n*}^T(K_{nn} + \sigma_n^2 I)^{-1} K_{n*})$$

while σ_n^2 is the variance of Gaussian likelihood, K_{nn} is the kernel matrix between training samples D_n and K_{n*} denotes the kernel matrix between the training samples and the test point.

Various AL approaches have been proposed to identify the feasible region (Bryan et al. 2006, Picheny et al. 2010, Chen and Fuge 2018, Knudde et al. 2019). In modern engineering simulation applications, the increasing complexity and high dimensionality of the problem necessitates the use of more data to accurately identify the feasible region. Despite the readily available computation resources, no feasible region identification algorithm has yet taken advantage of running simulations in parallel, that is, using batch active learning. Though some approaches have been proposed to tackle this problem, most contributions are restricted to the Bayesian optimization scope. Perhaps the most related approach is Gotovos et al. (2013), but their AL strategy is built on choosing samples from an existing set. AL for feasible region identification in a batch setting is still very much an open issue.

Contributions We contribute a new framework to identify the feasible region using AL in a parallel setting. We notice that common active feasible region identification techniques try to sample near the feasible boundary. With this intuition, we follow the approach of González et al. (2016) to leverage the general assumption (Alvi et al. 2019, Kim and Choi 2019, González et al. 2016, Malherbe and Vayatis 2017) that f is a Lipschitz continuous function. As conveyed in Figure 1, by imposing an upper bound on the function gradient we develop a local penalization function to take the effects of the already selected batch samples into account. We extend several state-of-the-art batch AL methods (Zhan et al. 2017, Contal et al. 2013) for feasible region identification. We conduct numerical experiments on these methods and

show better performance of our method. The proposed batch AL method is developed for the entropy acquisition function of Knudde et al. 2019, however, it can also be used with other acquisition functions for feasible region identification.

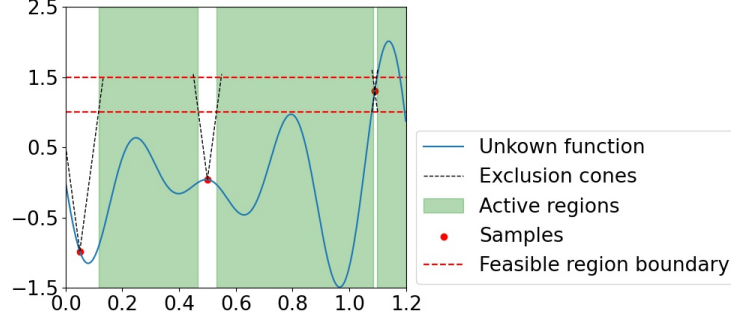


Figure 1: A 1 dimensional demonstration of batch AL to characterize the feasible region $A := \{x \in \mathcal{X} : 1 \leq f(x) \leq 1.5\}$. By assuming the unknown function f is Lipschitz continuous, we can leverage the Lipschitz constant based exclusion cones to determine some *exclusion regions* (white area) centered at the three data points (red dots). Within these regions there is no chance for the function to pass any of the feasible boundaries without violating the Lipschitz continuity assumption. In a batch setting, these exclusion regions are incorporated into the acquisition function to avoid further sampling close to already selected batch samples.

The remainder of this paper is organized as follows: Section 2 provides a review on active learning for feasible region identification, as well as discusses existing approaches for batch active learning for regression problems. In Section 3, a novel batch active feasible region identification method is presented. The performance of the proposed methodology is benchmarked with a state-of-the-art acquisition function in Section 4. Section 5 concludes the paper.

2 RELATED WORK

2.1 Acquisition Functions For Feasible Region Identification

The core of model-based Bayesian active learning is the acquisition function, which quantifies how informative a new candidate sample might be. Picheny et al. (2010) propose a weighted integration mean square error strategy to measure the metamodel’s belief about the accuracy of the feasible region. Based on this measure a new sampling technique is developed by leveraging the fact that model variance is not affected by the function response at new data. In a classification setting, Houlsby et al. (2012) proposed Bayesian Active Learning by Disagreement (BALD) method, and demonstrate how it can be employed to actively learn a Gaussian Process classifier. A variant of the Probability of Feasibility (PoF) (Forrester et al. 2008) has been proposed by Kaintura et al. (2018), where the variance has been heuristically added to encourage exploring within the feasible region. To find the feasible region in an unbounded design space, an active expansion sampling technique is introduced by Chen and Fuge (2018). Recently, an information-based acquisition function (Knudde et al. 2019) has also been proposed and demonstrates state-of-the-art performance compared with other existing approaches according to numerical experiments.

The acquisition function proposed by Knudde et al. (2019) represents the loss in entropy of the posterior distribution of interest g_i :

$$\alpha(\mathbf{x}) = \sum_{i=1}^k [\mathbb{H}(p(g_i|D_n)) - \mathbb{E}_{p(y|D_n, \mathbf{x})}(\mathbb{H}(p(g_i|D_n \cup \{\mathbf{x}, y\})))] \quad (1)$$

where k denotes the output space partitioned by the feasible boundaries. For instance, a feasible region defined on a bounded interval $[\alpha, \beta]$ will partition the output space into 3 different categories: $g_1 : y < \alpha$, $g_2 : \alpha \leq y < \beta$ and $g_3 : \beta \leq y$. One can employ the trick of conditional mutual information (Houlsby et al. 2011) to calculate Eq.(1), which then has a closed form:

$$\alpha(\mathbf{x}) = \sum_{i=1}^k \left[-\ln Z_i - \frac{1}{2Z_i} (\alpha_i - \mu(\mathbf{x})) \mathcal{N}(\mu(\mathbf{x}) | \alpha_i, \sigma^2(\mathbf{x})) - (\beta_i - \mu(\mathbf{x})) \mathcal{N}(\mu(\mathbf{x}) | \beta_i, \sigma^2(\mathbf{x})) \right]$$

where $Z_i = \Phi(\frac{\beta_i - \mu(\mathbf{x})}{\sigma(\mathbf{x})}) - \Phi(\frac{\alpha_i - \mu(\mathbf{x})}{\sigma(\mathbf{x})})$, and $\Phi(\cdot)$ is the standard normal cumulative density function.

This acquisition function has shown state-of-the-art performance compared with other methods. Hence without loss of generality, in this work we employ it for the numerical experiments. We refer to Knudde et al. (2019) for experiment details and more in-depth discussions.

2.2 Batch Bayesian Active Learning

The goal of a batch acquisition function is to imitate the future decisions that would be made under the equivalent (optimal) sequential policy (González et al. 2016). However, this needs to take already selected batch samples and their possible outcomes into consideration, which results in the following marginalization:

$$\mathbf{x}_{t,q} = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \int \alpha(\mathbf{x}; I_{t,q-1}) \prod_{j=1}^{q-1} p(y_{t,j} | \mathbf{x}_{t,j}, I_{t,j-1}) p(\mathbf{x}_{t,j} | I_{t,j-1}) d\mathbf{x}_{t,j} dy_{t,j} \quad (2)$$

$\alpha(\mathbf{x}; I_{t,q-1})$ is the acquisition function in batch iteration t , step q , where $q-1$ samples are already selected for this batch. $I_{t,q-1}$ represents the available information so far: training data D_n and the GP posterior in current batch iteration t . The outcome distribution $p(y_{t,j} | \mathbf{x}_{t,j}, I_{t,j-1})$ is Gaussian distributed:

$$p(y_{t,j} | \mathbf{x}_{t,j}, I_{t,j-1}) = \mathcal{N}(y_{t,j}; \mu(\mathbf{x}_{t,j}), \sigma^2(\mathbf{x}_{t,j}))$$

$p(\mathbf{x}_{t,j} | I_{t,j-1}) = \delta(\mathbf{x}_{t,j} - \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \alpha(\mathbf{x}; I_{t,j-1}))$ is the distribution of \mathbf{x} in step j , batch t , which, in practice, is obtained by maximization of the acquisition function $\alpha(\mathbf{x}; I_{t,j-1})$.

The first batch sample $\mathbf{x}_{t,1}$ can be trivially obtained by an *off-the-shelf* acquisition function optimization process. Unfortunately, it requires unaffordable computation effort for discovering the remaining $q-1$ samples of the batch, which is difficult to handle even for small batch sizes. To address this issue, different approaches have been proposed.

In BO, one common strategy is to leverage the fact that generally acquisition functions try to sample at global optima. With this purpose, Contal et al. (2013) defines a *relevant region*: $\mathcal{R}_t \subset \mathcal{X}$ in the input space where the unknown function has a higher possibility to contain optima. The remaining $q-1$ samples are then acquired by conducting pure exploration to reduce the uncertainty of model prediction within those relevant regions. This technique proves to be equivalent to maximizing the determinant to select a fixed cardinality subset via Determinantal Point Processes (DPP) (Kathuria et al. 2016), also known as k -DPP (Kulesza and Taskar 2011). Alternatively a MCMC sampling method can also be employed to sample the fixed-subset from DPP within the same defined relevant region \mathcal{R}_t and shows preferable performance.

González et al. (2016) assume the function is Lipschitz continuous. This assumption allows placing upper bounds on the function gradient, which leads to exclusion regions based on pre-visited batch samples, where function has no chance to become global optimum. These regions can be safely skipped in the acquisition function optimization. For high dimensional problems, Wang et al. (2018) propose to use a Mondrian processes to randomly partition the design space and, subsequently, perform BO based on each of these spaces. The batch samples are then selected according to a metric considering both diversity and quality. To avoid sampling too close within a batch, a penalty function constructed by using the kernel

estimated correlation is proposed by Zhan et al. (2017). Recently, the BALD method for classification has also been extended by Kirsch et al. (2019) to a batch setting.

Besides mimicking the sequential policy by exploiting properties of the AL goals. Some approaches have also been proposed that leverage the joint distribution of $y_{t,1}, \dots, y_{t,q}$ from GP prediction to form the batch acquisition function (Chevalier and Ginsbourger 2013, Shah and Ghahramani 2015). For instance, sampling a batch of points to get the maximum expected improvements (Chevalier and Ginsbourger 2013, Wang et al. 2016). However, these non-greedy strategies tend to scale poorly with the dimension of the problem and the size of the batches (González et al. 2016).

3 BATCH BAYESIAN ACTIVE FEASIBLE REGION IDENTIFICATION

The main idea of the proposed Batch active Feasible Region Identification by Local Penalization (BFRI-LP) method is assuming the dominant effect of a function evaluation on the acquisition function is a local exclusion around the new evaluation (González et al. 2016), see Figure 2. This behavior can be frequently seen in empirical experiments and is especially common when the acquisition function is multi-modal. Given this intuition, we can approximate Eq.(2) by formulating a factorized penalization function $\phi(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. The penalization function is monotonically increasing with respect to the Euclidean distance of a test point \mathbf{x} and an existing sample $\mathbf{x}_{t,j}$, thus imposing a local effect on the acquisition function at pre-selected samples in a batch:

$$\mathbf{x}_{t,q} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \left\{ g(\alpha(\mathbf{x}; I_{t,0})) \prod_{j=1}^{q-1} \phi(\mathbf{x}; \mathbf{x}_{t,j}) \right\}$$

where $g(\alpha) = \ln(1 + e^\alpha)$ applies a *soft-plus* transformation to ensure the acquisition function $\alpha(\cdot)$ is non-negative.

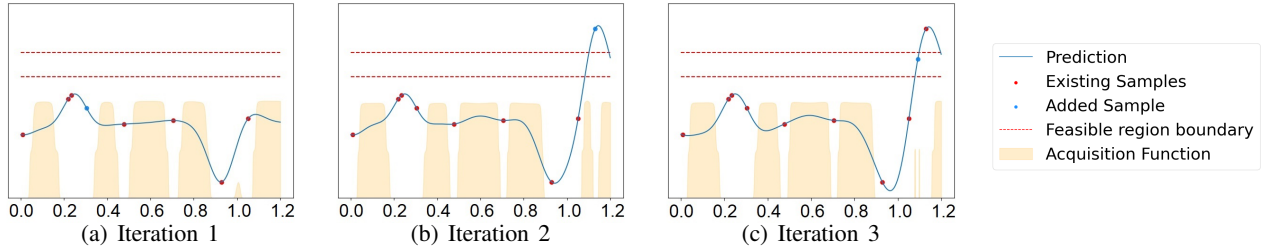


Figure 2: A typical active learning process for feasible region identification using the acquisition function from Knudde et al. (2019). New samples are added sequentially through maximization of the acquisition function. It can be seen that the main effect of adding a new sample is a decrease in the acquisition function as model uncertainty decreases at that location.

In practice, the penalty function needs to be carefully chosen as it might otherwise exclude too large areas of the original acquisition function with the risk of neglecting some interesting candidates. In BO, these interesting candidates are optima of the unknown function. In our case, we argue its where the outcome of the unknown function is exactly at the feasible boundary. This intuition comes naturally as no more information can be retrieved than finding the decision boundary for a classification problem. Moreover, this assumption can also be empirically observed (Knudde et al. 2019 Chen and Fuge 2018, Bryan et al. 2006) in active learning for feasible region identification. Under this assumption, we develop a probabilistic local penalization function ϕ in the following section. It will remain possible for the upcoming samples to be added at those interesting candidates.

3.1 A Local Penalization Function

In order to leverage the intuition that more information exists at the feasible boundary, one general approach to formulate such a local penalty function is to utilize the Lipschitz continuity of the unknown function. With the assumption that there is an upper bound to the function gradient, we can formulate a local penalization function, see in Figure 1. Specifically, considering the following ball-shape region in the Euclidean design space \mathcal{X} :

$$B_{r_j}(\mathbf{x}) = \left\{ \mathbf{x} \in \mathcal{X} : \|\mathbf{x}_j - \mathbf{x}\| \leq \left\lfloor \frac{c - f(\mathbf{x}_j)}{L} \right\rfloor \right\}$$

where c is the feasible boundary pre-specified by the users, L is the Lipschitz constant, and f is the unknown function. The ball-shaped region centered at \mathbf{x}_j provides an exclusion region in the design space where it is less likely for the unknown function f passing through the feasible boundary c .

Of course, in batch AL the outcome $f(\mathbf{x}_j)$ is unknown and modeled by the GP posterior. Thus the ball-shaped region is also stochastic. Meanwhile, as shown in Figure 1 and Figure 2, the feasible region is defined on a bounded interval $[\alpha, \beta]$ in the output space. Hence more than one feasible boundary exist. For this case, we are interested in the intersection of all ball-shaped regions $B_{r_{j_1}}, B_{r_{j_2}}$ so that the function will not pass through any of the boundaries.

The penalty function can thus be defined as the probability that for any $\mathbf{x} \in \mathcal{X}$, it might become a candidate sample that located at any feasible boundary.

$$\begin{aligned} \phi(\mathbf{x}; \mathbf{x}_j) &= 1 - p(\mathbf{x} \in B_{r_{j_1}}, \mathbf{x} \in B_{r_{j_2}}) \\ &= 1 - p\left(\|\mathbf{x}_j - \mathbf{x}\| \leq \left\lfloor \frac{\alpha - f(\mathbf{x}_j)}{L} \right\rfloor, \|\mathbf{x}_j - \mathbf{x}\| \leq \left\lfloor \frac{\beta - f(\mathbf{x}_j)}{L} \right\rfloor\right) \\ &= \varphi\left(\frac{L\|\mathbf{x}_j - \mathbf{x}\| + \beta - \mu(\mathbf{x}_j)}{\sigma(\mathbf{x}_j)}\right) - \varphi\left(\frac{\alpha - L\|\mathbf{x}_j - \mathbf{x}\| - \mu(\mathbf{x}_j)}{\sigma(\mathbf{x}_j)}\right) + \\ &\quad \max\left(\varphi\left(\frac{\beta - L\|\mathbf{x}_j - \mathbf{x}\| - \mu(\mathbf{x}_j)}{\sigma(\mathbf{x}_j)}\right) - \varphi\left(\frac{\alpha + L\|\mathbf{x}_j - \mathbf{x}\| - \mu(\mathbf{x}_j)}{\sigma(\mathbf{x}_j)}\right), 0\right) \end{aligned}$$

where φ is the probability density function of the standard Gaussian distribution. The complete BFRI-LP strategy is outlined in Algorithm 1.

3.2 Extending Batch AL methods For Feasible Region Identification

Besides the proposal of BFRI-LP, we also extend some current state-of-the-art batch methods (Contal et al. 2013, Zhan et al. 2017) for the feasible region identification problem. For the pure-exploration method of Contal et al. 2013, the relevant region \mathcal{R}_t is redefined to locate feasible boundaries.

$$R_t := R_1 \cup R_2$$

where

$$\begin{aligned} R_1 &:= \{\mathbf{x} \in \mathcal{X} : \mu(\mathbf{x}) + 1.96\sigma(\mathbf{x}) \geq \alpha \\ &\quad \text{or } \mu(\mathbf{x}) - 1.96\sigma(\mathbf{x}) \leq \alpha\} \\ R_2 &:= \{\mathbf{x} \in \mathcal{X} : \mu(\mathbf{x}) + 1.96\sigma(\mathbf{x}) \geq \beta \\ &\quad \text{or } \mu(\mathbf{x}) - 1.96\sigma(\mathbf{x}) \leq \beta\} \end{aligned}$$

The influence function method (Zhan et al. 2017) is more directly applicable by adding a soft-plus transformation to the acquisition function.

Algorithm 1: Batch Bayesian active Feasible Region Identification with Local Penalization (BFRI-LP)

Input: dataset $D_n = \{\mathbf{X}_n, \mathbf{y}_n\}$, batch size Q , iteration budget N , acquisition transformation g , feasible boundary $[\alpha, \beta]$;

for $t := 1$ **to** N **do**

Evaluating the posterior of the GP: $I_{t,0} \leftarrow \{p(f|D_n)\}$;

Approximate the Lipschitz constant: $L \leftarrow \max_{\mathbf{x} \in \mathcal{X}} \|\mu_{\nabla}(\mathbf{x})\|$;

for $q := 1$ **to** Q **do**

if $q == 1$ **then**

$\mathbf{x}_{t,1} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}, I_{t,0})$;

else

$\mathbf{x}_{t,q} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} g(\alpha(\mathbf{x}, I_{t,0})) \prod_{j=1}^{q-1} \phi(\mathbf{x}, x_j)$;

end

end

Parallel query: $\mathbf{y}_{t,Q} = f(\mathbf{X}_{t,Q})$;

Expand dataset: $D_n : D_n \cup \{\mathbf{X}_{t,Q}, \mathbf{y}_{t,Q}\}$;

end

Returns Final GP Model for Prediction: $p(f|D_n)$;

4 NUMERICAL ANALYSIS

We conduct numerical experiments for various batch AL methods to investigate the performance.

4.1 Experimental Setup

We compare BFRI-LP with different existing approaches for batch AL. The first sample of the batch is chosen by the entropy method from Knudde et al. 2019, for feasible region identification, while the remaining batch samples are chosen using a standard batch AL method. The random sampling method, pure-exploration method (Contal et al. 2013), and the influence function method (Zhan et al. 2017) are included in the comparison and are referred to as, respectively, Entropy-Random, Entropy-PE, and Entropy-IF. To measure the effectiveness of batch AL, the one-by-one sequential sampling is also included and referred to as Entropy-Single in the experiments.

The benchmark functions (Chen and Fuge 2018, Picheny et al. 2013, Molga and Smutnicki 2005, Dette and Pepelyshev 2010, Adjiman et al. 1998, Sonja and Derek 2013) and relevant settings are provided in Table 1. For most cases the benchmark functions provide deterministic outcomes. However, we also include a stochastic benchmark based on Adjiman function with noisy observations, where we expect the Gaussian likelihood of the GP to tackle the noise and provide an accurate noise-free approximation. The initial data are provided by maximin Latin hypercube sampling. The benchmark functions start with 10 data points, except for Hartmann 6D and Dette 8D function where 30 data points are used.

For the batch AL methods, a batch size of 3 is used for all benchmark functions, except for Hartmann 6D and Dette 8D where the batch size is set to 5. In order to measure the final prediction accuracy, we use 100000 randomly distributed samples in the bounded design space as test data to calculate the F1 score. The code is implemented using GPFLOWOPT (Knudde et al. 2017). For the model training, the common Radial Basis Function (RBF) kernel is used with the automatic relevance detection (ARD) enabled. The GP model is re-trained 5 times during each batch iteration by an L-BFGS-B optimizer and the model with the best likelihood will be chosen to make predictions. For the acquisition function optimization, Monte Carlo sampling is first employed to provide a good initial candidate sample from 1000 random samples,

Table 1: Benchmark Function Settings

Function	Design Space	Feasible Boundaries $[\alpha, \beta]$	$V(A) \setminus V(\mathcal{X})(\%)$
Hosaki	$[0, 10]^2$	$[-\infty, -1]$	7.64
Branin-Hoo	$[0, 1]^2$	$[20, 40]$	24.68
Goldstein-Price	$[-2, 2]^2$	$[-\infty, 500]$	15.21
Curved Function	$[0, 1]^3$	$[1, 2]$	4.74
Six-hump camel	$[-3, 3] \times [-2, 2]$	$[-\infty, -0.1]$	5.57
Adjiman+ $\mathcal{N}(0, 0.2^2)$	$[-5, 5]^2$	$[-3, -2]$	3.71
Rosenbrock 4D	$[-2, 2]^4$	$[500, \infty]$	74.92
Piston Simulation (Scaled)	$[0, 1]^7$	$[0.4, 0.46]$	17.70
Hartmann 3D	$[0, 1]^3$	$[-\infty, -1.5]$	8.95
Hartmann 6D	$[0, 1]^6$	$[-\infty, -1.5]$	2.13
Dette 8D	$[0, 1]^8$	$[55, \infty]$	13.11

then this candidate point is set as the starting point of an L-BFGS-B gradient based optimizer to provide the final optimum of acquisition function. The gradient of the acquisition function and the posterior mean ($\mu_{\nabla}(\mathbf{x})$ of Algorithm.1) are obtained by automatic differentiation of Tensorflow (Abadi et al. 2016). Each benchmark configuration is repeated 10 times for statistical consistency.

4.2 Results And Discussion

In Figure 3, we highlight 6 batch iterations of the AL process of the Branin-Hoo function in detail to show how the penalty function influences the acquisition function. It can be seen that the GP prediction becomes much more accurate within 5 batch iterations. As the acquisition function keeps focusing near feasible boundaries, the penalization effect automatically shrinks to a very small region when the model becomes very certain of the feasible region.

The results of different feasible region identification methods are summarized in Figure 4. The performance metric for feasible region identification is the accuracy of the feasibility evaluation after applying the indicator function (F1 score) with respect to the amount of batch iterations. It can be seen that by exploiting parallel computation, all the batch methods can reach the same performance as the one-by-one sequential AL method with much less time. Moreover, BFRI-LP method outperforms the rest of the strategies. Interestingly, for the Branin-Hoo function the random sampling strategy can reach better results for some cases. This could be explained as the true feasible region covers a large region within the design space and is easier to sample. However, once the unknown function is more difficult to approximate and the feasible region is smaller, the advantage of BFRI-LP is more apparent. Entropy-PE outperforms BFRI-LP for a while in the noisy Adjiman case, though generally BFRI-LP shows more robust performance.

Finally, two experiments with varies batch sizes are conducted and the results are provided in Figure 5, it can be seen that the feasibility prediction accuracy generally increase faster with an increment of batch sizes.

5 CONCLUSION

We developed a data-efficient Bayesian technique for feasible region identification that is capable of utilizing all computation nodes using batch AL. This is done by building a penalization function based on the Lipschitz condition of the unknown function. Synthetic tests are carried out and it shows better performance over other relevant approaches. Future work will focus on extending the proposed strategy to handle multiple design constraints, and look at asynchronous batch AL where the computation time between samples varies.

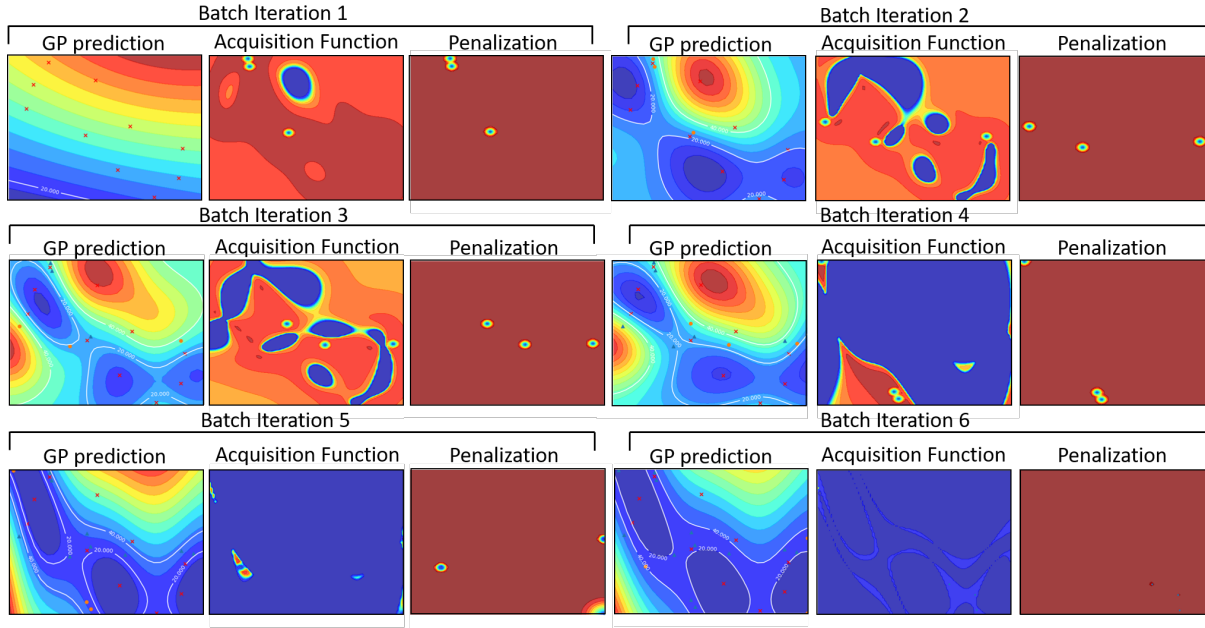


Figure 3: BFRi-LP on the Branin-Hoo function. For each batch iteration, the penalized acquisition function is maximized to sequentially produce one batch of samples. For brevity we only show the penalized acquisition function, as well as the penalization function itself, after selecting the complete batch. The red colored areas in the acquisition function and penalization function plots are of interest. The initial training samples are denoted as red crosses, the selected samples of previous batch iteration are denoted as orange dots, while evaluated samples of previous iterations are represented by blue triangles.

ACKNOWLEDGMENTS

This work is partially supported by the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme, and Chinese Scholarship Council under the grant number 201906290032.

REFERENCES

- Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin et al. 2016. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”. *arXiv preprint arXiv:1603.04467*. <https://arxiv.org/abs/1603.04467>, accessed 16th March 2016.
- Adjiman, C. S., S. Dallwig, C. A. Floudas, and A. Neumaier. 1998. “A Global Optimization Method, α BB, for General Twice-Differentiable Constrained NLPs—I. Theoretical Advances”. *Computers & Chemical Engineering* 22(9):1137–1158.
- Alvi, A. S., B. Ru, J. Calliess, S. J. Roberts, and M. A. Osborne. 2019. “Asynchronous batch Bayesian optimisation with improved local penalisation”. *arXiv preprint arXiv:1901.10452*. <https://arxiv.org/abs/1901.10452> accessed May 27th 2019.
- Azzimonti, D., D. Ginsbourger, C. Chevalier, J. Bect, and Y. Richet. 2019. “Adaptive Design of Experiments for Conservative Estimation of Excursion Sets”. *Technometrics* 0(0):1–14.
- Bryan, B., R. C. Nichol, C. R. Genovese, J. Schneider, C. J. Miller, and L. Wasserman. 2006. “Active Learning for Identifying Function Threshold Boundaries”. In *Advances in neural information processing systems*. Dec 4th-7th, Vancouver, B.C., Canada, 163–170.
- Chen, W., and M. Fuge. 2018. “Active Expansion Sampling for Learning Feasible Domains in an Unbounded Input Space”. *Structural and Multidisciplinary Optimization* 57(3):925–945.
- Chevalier, C., and D. Ginsbourger. 2013. “Fast Computation of the Multi-Points Expected Improvement with Applications in Batch Selection”. In *International Conference on Learning and Intelligent Optimization*. Springer. Jan 7th-11th, Catania, Italy, 59–69.

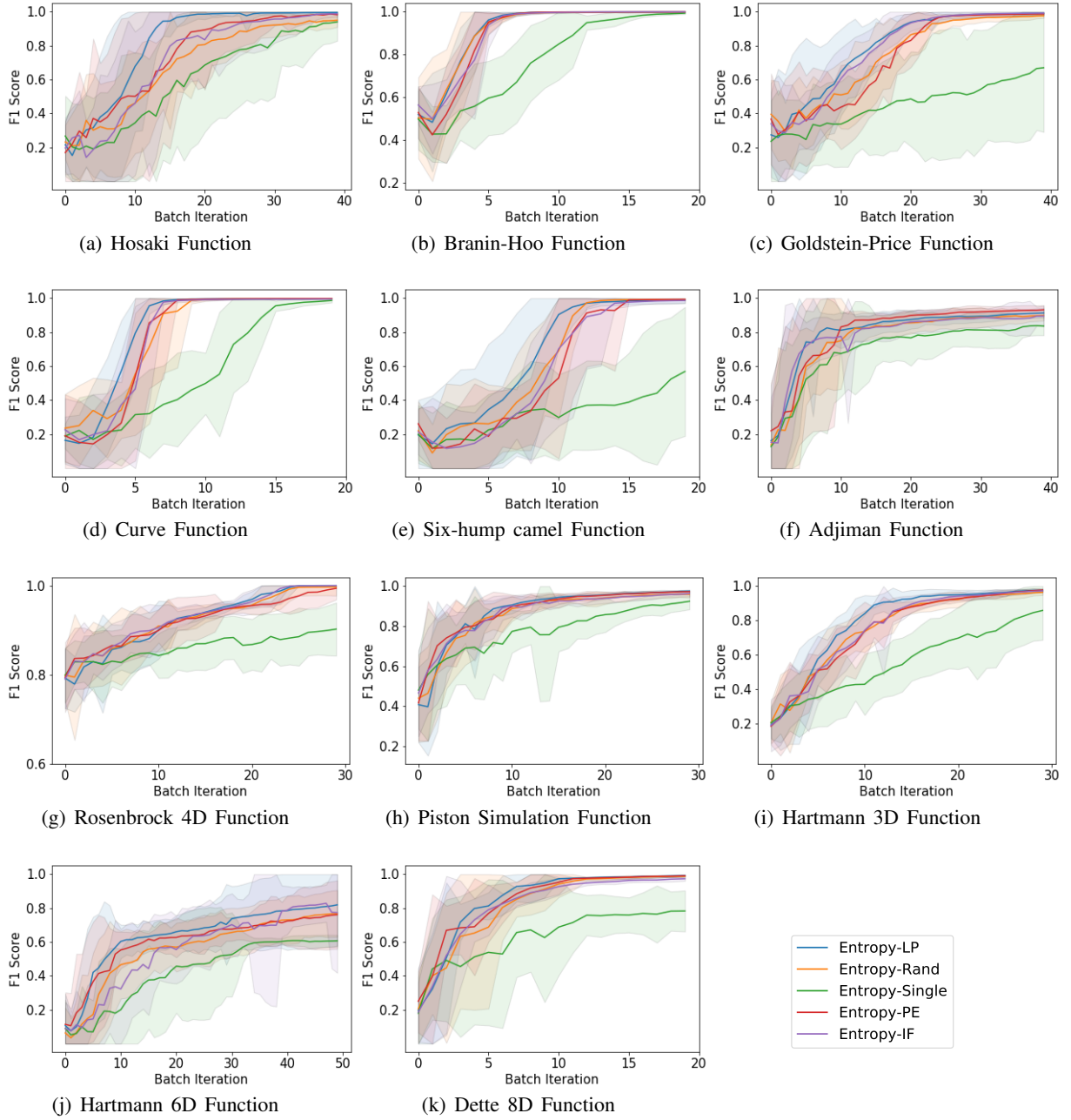


Figure 4: Results for the benchmark functions described in Table 1. It shows the evolution of the F1 score versus the batch iterations. The initial number of samples is 10, and samples are selected with a batch size of $q = 3$, except for Hartmann 6D and Dette 8D the initial number of samples are 30, and the batch size is $q = 5$.

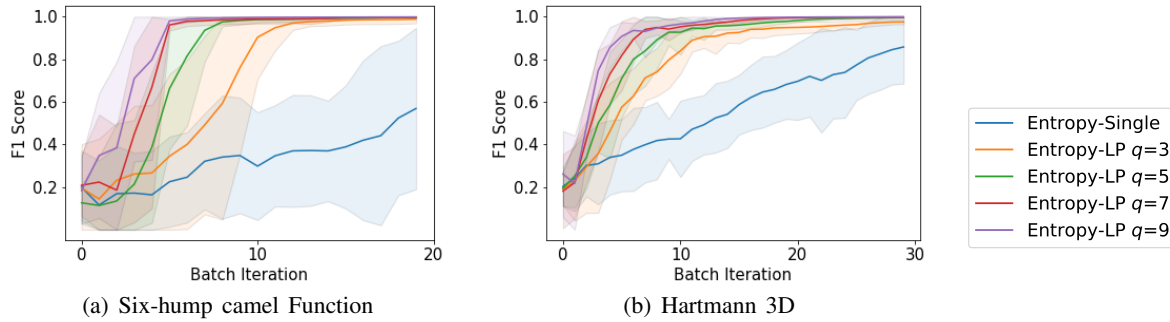


Figure 5: Performance of BFRI-LP on varying batch size q .

- Contal, E., D. Buffoni, A. Robicquet, and N. Vayatis. 2013. “Parallel Gaussian Process Optimization with Upper Confidence Bound and Pure Exploration”. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. Sep 23th-27th, Prague, Czech Republic, 225–240.
- Detle, H., and A. Pepelyshev. 2010. “Generalized Latin Hypercube Design for Computer Experiments”. *Technometrics* 52(4):421–429.
- Dubourg, V., B. Sudret, and F. Deheeger. 2013. “Metamodel-Based Importance Sampling for Structural Reliability Analysis”. *Probabilistic Engineering Mechanics* 33:47–57.
- Forrester, A., A. Sobester, and A. Keane. 2008. *Engineering Design via Surrogate Modelling: a Practical Guide*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Gelbart, M. A. 2015. *Constrained Bayesian Optimization and Applications*. Ph. D. thesis. Graduate School of Arts & Sciences, Harvard University, Cambridge, MA. <https://dash.harvard.edu/handle/1/17467236>, accessed 8th September.
- González, J., Z. Dai, P. Hennig, and N. Lawrence. 2016. “Batch Bayesian Optimization via Local Penalization”. In *Artificial intelligence and statistics*. May 9th-11th, Cadiz, Spain, 648–657.
- Gotovos, A., N. Casati, G. Hitz, and A. Krause. 2013. “Active Learning for Level Set Estimation”. In *Twenty-Third International Joint Conference on Artificial Intelligence*. Aug 3rd-9th, Beijing, China, 1344–1350.
- Hernández-Lobato, J. M., M. W. Hoffman, and Z. Ghahramani. 2014. “Predictive Entropy Search for Efficient Global Optimization of Black-box Functions”. In *Advances in neural information processing systems*. Dec 8th-13th, Montreal, Canada, 918–926.
- Houlsby, N., F. Huszar, Z. Ghahramani, and J. M. Hernández-Lobato. 2012. “Collaborative Gaussian processes for Preference Learning”. In *Advances in neural information processing systems*. Dec 3rd-8th, Nevada, USA, 2096–2104.
- Houlsby, N., F. Huszar, Z. Ghahramani, and M. Lengyel. 2011. “Bayesian Active Learning for Classification and Preference Learning”. *arXiv preprint arXiv:1112.5745*. <https://arxiv.org/abs/1112.5745>, accessed 24th December 2011.
- Kaintura, A., K. Foss, I. Couckuyt, T. Dhaene, O. Zografos, A. Vaysset, and B. Sorée. 2018. “Machine Learning for Fast Characterization of Magnetic Logic Devices”. In *2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*. Dec 16th-18th, Chhattisgarh, India, 1–3.
- Kathuria, T., A. Deshpande, and P. Kohli. 2016. “Batched Gaussian Process Bandit Optimization via Determinantal Point Processes”. In *Advances in Neural Information Processing Systems*. Dec 5th-10th, Barcelona, Spain, 4206–4214.
- Kim, J., and S. Choi. 2019. “On Local Optimizers of Acquisition Functions in Bayesian Optimization”. *arXiv preprint arXiv:1901.08350*. <https://arxiv.org/abs/1901.08350>, accessed 16th June 2020.
- Kirsch, A., J. van Amersfoort, and Y. Gal. 2019. “Batchbald: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning”. In *Advances in Neural Information Processing Systems*. Dec 8th-14th, Vancouver, Canada, 7024–7035.
- Knudde, N., I. Couckuyt, K. Shintani, and T. Dhaene. 2019. “Active Learning for Feasible Region Discovery”. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. Dec 16th-19th, Florida, USA, 567–572.
- Knudde, N., J. van der Herten, T. Dhaene, and I. Couckuyt. 2017. “GPflowOpt: A Bayesian optimization library using tensorflow”. *arXiv preprint arXiv:1711.03845*. <https://arxiv.org/abs/1711.03845>, accessed 10th November 2017.
- Kulesza, A., and B. Taskar. 2011. “k-DPPs: Fixed-Size Determinantal Point Processes”. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress. Jun 28th-July 2nd Washington, USA, 1193–1200.
- Larson, B. J., and C. A. Mattson. 2012. “Design Space Exploration for Quantifying a System Model’s Feasible Domain”. *Journal of Mechanical Design* 134(4).
- Malherbe, C., and N. Vayatis. 2017. “Global Optimization of Lipschitz Functions”. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. Aug 6th-11th, Sydney, Australia, 2314–2323.
- Molga, M., and C. Smutnicki. 2005. “Test Functions for Optimization Needs”.

- Picheny, V., D. Ginsbourger, O. Roustant, R. T. Haftka, and N.-H. Kim. 2010. "Adaptive Designs of Experiments for Accurate Approximation of a Target Region". *Journal of Mechanical Design* 132(7).
- Picheny, V., T. Wagner, and D. Ginsbourger. 2013. "A Benchmark of Kriging-based Infill Criteria for Noisy Optimization". *Structural and Multidisciplinary Optimization* 48(3):607–626.
- Rasmussen, C. E. 2003. "Gaussian Processes in Machine Learning". In *Summer School on Machine Learning*, 63–71. Springer.
- Rojas-Gonzalez, S., J. Branke, and I. Van Nieuwehuysen. 2019. "Multiobjective Ranking and Selection with Correlation and Heteroscedastic Noise". In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 3392–3403. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Shah, A., and Z. Ghahramani. 2015. "Parallel Predictive Entropy Search for Batch Global Optimization of Expensive Objective Functions". In *Advances in Neural Information Processing Systems*. Dec 7th-12th, Montreal, Canada, 3330–3338.
- Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. 2015. "Taking the Human Out of The Loop: A Review of Bayesian Optimization". *Proceedings of the IEEE* 104(1):148–175.
- Snoek, J. R. 2013. *Bayesian Optimization and Semiparametric Models with Applications to Assistive Technology*. Ph. D. thesis. <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.719.5070&rank=1> accessed 8th September.
- Sonja, S., and B. Derek. 2013. "Virtual Library of Simulation Experiments: Test Functions and Datasets". <http://www.sfu.ca/~ssurjano>, accessed 22nd June 2020.
- Van Steenkiste, T., J. van der Herten, I. Couckuyt, and T. Dhaene. 2016. "Sensitivity Analysis of Expensive Black-Box Systems Using Metamodeling". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 578–589. Washington, USA: Institute of Electrical and Electronics Engineers, Inc.
- Wang, J., S. C. Clark, E. Liu, and P. I. Frazier. 2016. "Parallel Bayesian Global Optimization of Expensive Functions". *arXiv preprint arXiv:1602.05149*. <https://arxiv.org/abs/1602.05149> accessed 5th May 2019.
- Wang, Z., C. Gehring, P. Kohli, and S. Jegelka. 2018, 09–11 Apr. "Batched Large-scale Bayesian Optimization in High-dimensional Spaces". Volume 84 of *Proceedings of Machine Learning Research*, 745–754. Playa Blanca, Lanzarote, Canary Islands: PMLR.
- Zhan, D., J. Qian, and Y. Cheng. 2017. "Pseudo Expected Improvement Criterion for Parallel EGO algorithm". *Journal of Global Optimization* 68(3):641–662.

AUTHOR BIOGRAPHIES

Jixiang Qing received his M.E. degree in Aerospace Engineering from Northwestern Polytechnical University in 2019. Starting from July 2019 he is active as a PhD student in the Internet Technology and Data Science Lab (IDLab) research group at Ghent University, working on Bayesian active learning, optimization and generative design. His e-mail address is Jixiang.Qing@ugent.be.

Nicolas Knudde received his M.Sc. degree in Engineering Physics from Ghent University in 2016. Starting from September 2016 he is active as a PhD student in the research group Internet Technology and Data Science Lab (IDLab) at Ghent University, working on Bayesian optimization and active learning. His e-mail address is Nicolas.Knudde@ugent.be.

Ivo Couckuyt obtained his PhD degree from Ghent University in 2013. He is currently working as a postdoctoral research fellow at Ghent University - IDLab. His research is mainly focused on global and local surrogate modeling (metamodeling) and its application to solve real world problems, optimization of expensive functions, evolutionary computing and machine learning methods. His e-mail address is ivo.couckuyt@ugent.be.

Tom dhaene is a full professor at the Department of Information Technology (INTEC) of Ghent University. He received the PhD degree in electrical engineering from Ghent University, Belgium, in 1993. In September 2000, he joined the Department of Mathematics and Computer Science of the University of Antwerp as a Professor. Since October 2007, he has been a Full Professor with the Department of Information Technology, Ghent University. He is also affiliated with imec. As author or co-author, he has contributed to more than 350 peer-reviewed papers and abstracts in international conference proceedings, journals and books about computational science and numerical analysis and engineering. He is the holder of five U.S. patents. His e-mail address is tom.dhaene@ugent.be.

Kohei Shintani received his PhD from Nagoya University. He works as an assistant manager in Toyota motor corporation. He is responsible for development of vehicle performance and optimization technology. His academic experience includes working as a visiting scholar in Northwestern University in 2016 and Nagoya University in 2019. His e-mail address is kohei_shintani@mail.toyota.co.jp.